MusalaSoft

Processing of big data with Apache Spark

JavaSkop ‘18

Aleksandar Donevski

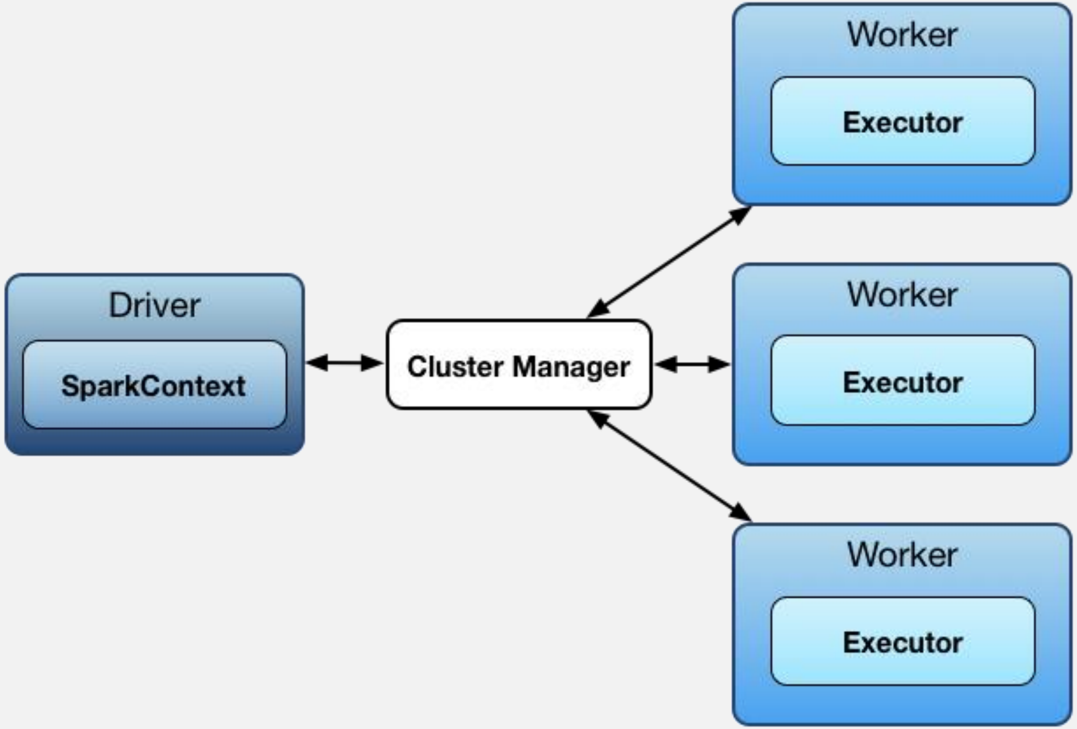
AGENDA

- **What is Apache Spark?**
- Spark vs Hadoop MapReduce
- Application Requirements
- Example Architecture
- Application Challenges

WHAT IS APACHE SPARK?

- Engine for processing of large-scale data
- Open source
- Interact with Java, Scala, Python, and R
- Run as Standalone or on YARN, Kubernetes, and Mesos
- Access HDFS, HBase, Cassandra, S3, and etc.

SPARK ARCHITECTURE



RESILIENT DISTRIBUTED DATASET (RDD)

- Characteristics:
  - **Immutable, distributed, partitioned, and resilient**

- API:
  1. Transformations:
     - map(), filter(), distinct(), union(), subtract(), and etc.

  2. Actions:
     - reduce(), collect(), count(), first(), take(), and etc.

RDD OPERATIONS

- **Transformations** are executed on **workers**

- **Actions** may transfer data from the workers to the **driver**
  - collect() sends all the partitions to the single driver

- Persistence:
  - persist() and cache()

AGENDA

- What is Apache Spark?
- **Spark vs Hadoop MapReduce**
- Application Requirements
- Example Architecture
- Application Challenges

SPARK VS MAPREDUCE

- Spark
  - Real time, streaming
  - Processes data in-memory
  - Handle structures which could not be decomposed to key-value pairs

- MapReduce
  - Batch mode, not real-time
  - Persist on disk after map operation
  - Key-value pairs

AGENDA

- What is Apache Spark?
- Spark vs Hadoop MapReduce
- **Application Requirements**
- Example Architecture
- Application Challenges

APPLICATION REQUIREMENTS

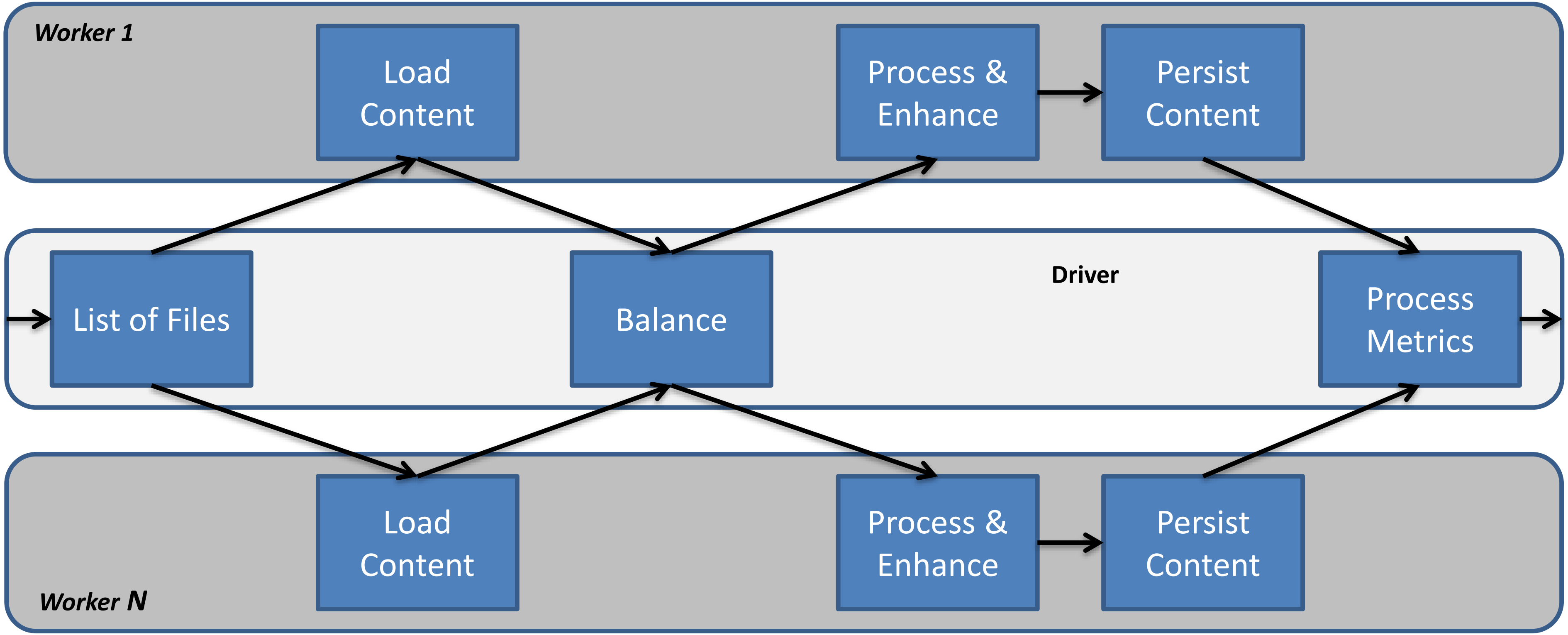
- Verify that application is thread-safe
- Use synchronization blocks appropriately
- Avoid duplication of objects
- Try to use array of objects and primitive types
- Avoid unneeded data in the objects
- *Always remember that application is executed in parallel!*

APPLICATION PIPELINE

- Define the application pipeline with usage of SparkContext object
- “Encapsulate” the common data needed through all pipeline steps
- Prepare the common data and broadcast it through the workers as needed

AGENDA

- What is Apache Spark?
- Spark vs Hadoop MapReduce
- Application Requirements
- **Example Architecture**
- Application Challenges


Worker 1
Load Content
Process & Enhance
Persist Content
List of Files
Balance
Driver
Process Metrics
Load Content
Process & Enhance
Persist Content
Worker N

AGENDA

- What is Apache Spark?
- Spark vs Hadoop MapReduce
- Application Requirements
- Example Architecture
- **Application Challenges**

DEPENDENCY ISSUES

- Example:
  - Log4j 1.x – Spark
  - Log4j 2.x – Application (Async Loggers)
  - Application fails at the very beginning

- Resolution:
  - Shading dependencies
  - Provide the dependencies in “--jars” property and add “spark.{driver,executor}.userClassPathFirst=true” properties

MEMORY ISSUES

- Example:
  - Default usage of 1Gb RAM per executor
  - Executors fail with OOM error, thus application fails

- Resolution:
  - Verify cluster available memory
  - Monitor and measure memory usage
  - Tune per application case

PERFORMANCE ISSUES

- Example:
    - Application execution time is taking too long for simple set of data
    - Last task executing time is taking too long

- Resolution:
    - Verify the partitioning
    - Adjust the processing time of each task

APPLICATION ISSUES

- Example:
  - Default Java serialization is being used
  - Serialization time is taking too long

- Resolution:
  - Verify the objects data and data structures used
  - Use Kryo serialization

API ISSUES

- Three to four month cycle releases
- Lots of hood changes
- Verification if application is affected